Kerberos
Defending Champion

A trusted third-party authentication service, Kerberos uses secret key cryptography to provide a centralised, strong authentication for client/server applications. In spite of a few limitations, it will set to rest most of your security worries.

Securing your network is very critical to business. It becomes even more challenging if your network is accessible to the outside world. It's your responsibility to make sure that your resources and confidential information are secured and protected from unauthorised access. The conventional way of authentication using a password is no longer considered foolproof, particularly in cases where the user's password is sent over the network in clear plain text. Any attacker (whether internal or external) who has access to the network can sniff out those passwords and cause a huge loss to your business by attacking your network in a variety of ways. Here I'll discuss how the Kerberos network authentication protocol can play a vital role in securing your network.

Basics of network security

Authentication, authorisation, accounting and secured communication are some of the key areas of network security. Authentication is the process of verifying the identity of a particular user. The most common form of authentication is through user ID and passwords. Authorisation is the process of granting or denying access to specific resources based on the user's identity. The most common form of authorisation is through access control lists. Accounting is the process of recording who accessed which resources. The most common form of accounting is through some form of logging into audit logs. Secure communication is the ability to protect the network transmissions from both interception and unauthorised transmission. The most common approach for secured communication is through encryption.

Kerberos network authentication protocol

Kerberos is a network authentication protocol from the Massachusetts Institute of Technology (MIT). It is designed to provide a centralised, strong authentication for client/server applications by using secret-key cryptography.

Kerberos solves three common network security problems: It allows network administrators to maintain a single, centralised password store; it prevents passwords from being intercepted on the network; and it frees users from repeatedly authenticating themselves to the application servers throughout the network.

The centralised password store eases the burden of the network administrator, as she now only needs to maintain a single system hosting the user name/password database and can specially harden and secure the system.

Kerberos never transmits passwords in clear text over the network. Instead, it uses strong cryptography to generate forgery-proof, time-limited digital authentication tickets and allows users to access the network resources by simply presenting these secure tickets rather than repeatedly entering user IDs and passwords.

Kerberos is not a complete network security solution. It only provides network-wide user authentication and secure communications. It has no provisions for authorisation and accounting. Although Kerberos itself includes a minimal access control list to specify users authorised to change the Kerberos database, no other authorisation system is provided. Likewise, in the Kerberos environment, as the actual authentication takes place on the workstation, the Kerberos server does not concern itself with who is successfully authenticated. While Kerberos does keep track of which tickets were issued, it provides more of a debugging function than an auditing function.

Kerberos components

Before seeing the various components involved in the Kerberos environment, you must have a clear understanding of the following terms:

User: A human who uses a program or service.

Client: Often a program that will contact the server on behalf of the user.

Server: A program that usually provides some sort of service to the client.

Principal: A user or a client or a server. As far as Kerberos is concerned, both the client and the server that uses the Kerberos service is a client. So to distinguish the Kerberos clients from other clients, the term 'Principal' is used. Principals are identified in the following notation:

user/instance@REALM

Instance: Allows multiple principals for a specific user, e.g. raja@DEMO.COM, raja/admin@DEMO.COM

Realm: An administrative unit serviced by a single Kerberos server-hosting database. It often reflects the DNS name.

Secret key: A large number assigned to the Kerberos principal. In the case of a user, it is derived from the user's password, and in the case of a server, it is a random number. Only Kerberos and the principal know the secret key.

Ticket: A record that helps a client to authenticate itself to the server. It usually contains the client's identity, server name and other information all encrypted using the server's secret key.

Access control list (ACL): This will contain a list of principals authorised to modify the Kerberos database.

Kerberised application: These are applications that are Kerberos-aware, e.g. Kerberised FTP servers/clients, Kerberised telnet servers/clients, etc.

There are at least three components involved in the authentication process in a given Kerberos realm: workstation, application server and the Kerberos server (Figure 1).

The workstation (or the client) is the one that runs Kerberised client applications, such as an FTP client or telnet client, on behalf of the user. The application server (or the server) hosts the Kerberised server applications. Like the FTP server, telnet server etc, the Kerberos server hosts the Key Distribution Center (KDC).

The KDC is an integral part of the Kerberos system. It consists of three logical components: a database of all principals and their associated secret keys, the authentication server (AS) and the ticket-granting server (TGS) (refer Figure 2).

The AS issues a ticket-granting ticket (TGT) to the clients who wish to use the services hosted by the applications servers. The TGT can be used to request individual service tickets. The TGS is the one that issues individual service tickets to the clients.

How does Kerberos work?

Let's consider a client that wants to connect to an application server using Kerberos.

First, the client requests a TGT to the AS running in the KDC (Figure 3).

If the AS finds the client principal in the Kerberos database, it creates a TGT and encrypts it with the TGS secret key. And it replies to the client giving the credentials consisting of the TGS session key and the TGT, both of which are encrypted with the client's secret key (Figure 4).

Upon receiving the credentials from the AS, the client decrypts the credentials with the client's secret key (derived from the client's password) to get the TGS session key. At the end of this one-time initial authentication process, the client has the TGT (encrypted with a TGS secret key) and a TGS session key. The client can use the TGT to request individual service tickets. The TGS session key can be used for secure communication between the client and the TGS server. This initial authentication can either happen automatically at the time of logging in to the system if the client uses the Kerberised login program, or manually by running the *kinit* program.

It should be noted that no one, other than the legitimate client, can decrypt the credentials received from the AS, as the client's secret key is known only to the client and the AS. Also, note that no password is sent over the network in clear text.

To access a particular service running on the application server, the client creates an authenticator encrypted with a TGS

Figure 1: The three components

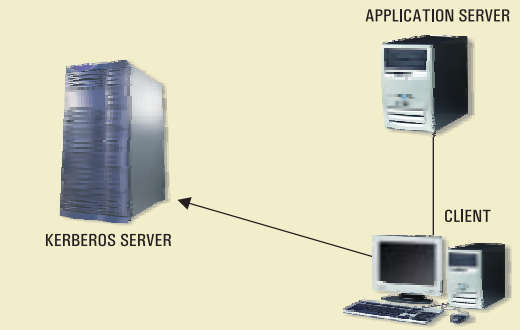


Figure 2: KDC's three logical components

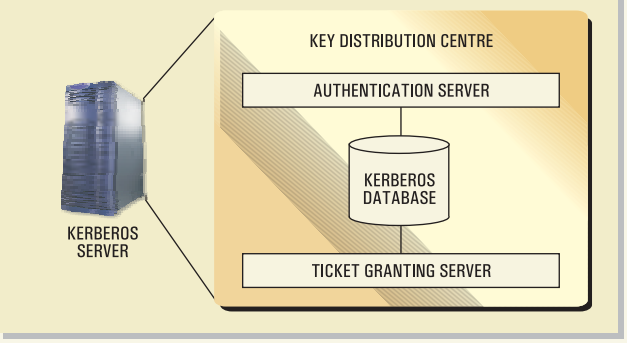


Figure 3: Client requests a TGT from AS

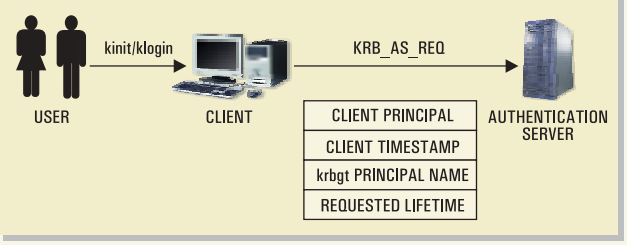


Figure 4: AS replies to the client

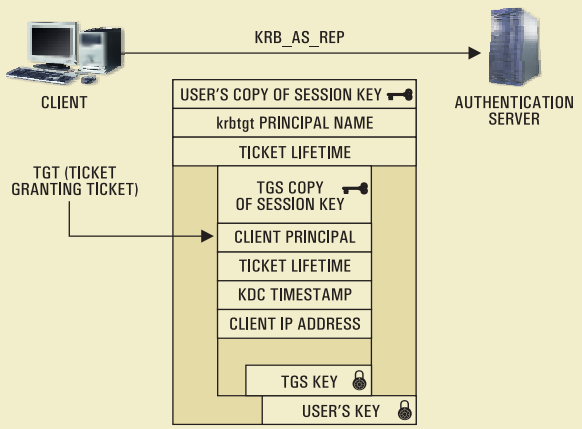


Figure 5: The client reacts

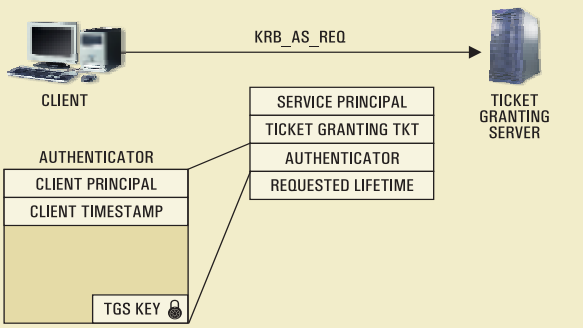


Figure 6: TGS creates the service ticket

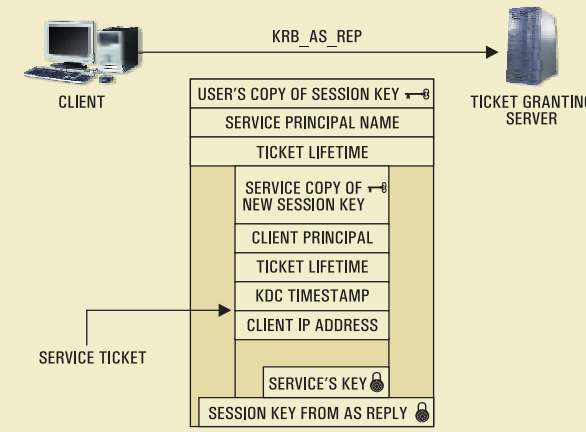
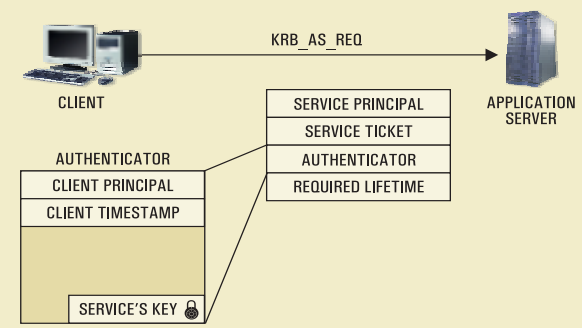


Figure 7: Client sends authenticator to AS



session key to prove its identity to the TGS. The client then sends the authenticator and the TGT to the TGS (Figure 5).

Upon receiving the request from the client, the TGS decrypts the TGT with its secret key to get the TGS session key. It then decrypts the authenticator with the TGS session key to verify the client's identity (as only the TGS and the client know the TGS session key and no one else can impersonate as the legitimate client). The TGS then creates the service ticket, encrypts it with the service secret key and then replies to the client, giving the credentials consisting of the service ticket and the service session key, both of which are encrypted with the TGS session key (Figure 6).

Upon receiving the credentials from the TGS, the client decrypts it with the TGS session key to get the service ticket and service session key. It then creates an authenticator—this time it is encrypted with a service session key. The client then sends the authenticator and service ticket to the service running on the application server (Figure 7).

Upon receiving the request, the service running on the application server decrypts the service ticket with its secret key to get the service session key. It then decrypts the authenticator with the service session key to verify the client's identity. If all goes well, the client is allowed to establish a connection with the service.

Limitations

Attacks like 'Denial Of Service' or 'Password Guessing' have not been solved in Kerberos. To a user, this means that an attacker can prevent Kerberos from serving genuine clients or can successfully break into the system by decrypting poorly chosen passwords.

If either the server hosting the Kerberos or the client/server principal is compromised, then the entire system is compromised.

Each client participating in the Kerberos authentication mechanism must have his system clocks synchronised, possibly using Network Time Protocol. If this is not done, then authentication becomes impossible and network services become unavailable.

Kerberos provides a secure and complex authentication service over insecure networks, and optionally encrypts communication between two end-points. It does all this without sending data across the network that could allow attackers to impersonate the legitimate client. It strives to improve security and convenience at the same time, and is sure to play a vital role in the security of your network. LFY

By: R.K. Raja. The author is a lead engineer working with HCL Technologies (Cisco Systems Offshore Development Centre) in Chennai. He can be contacted at: rajark_hcl@yahoo.co.in